

Docket No. P2092D/1612US

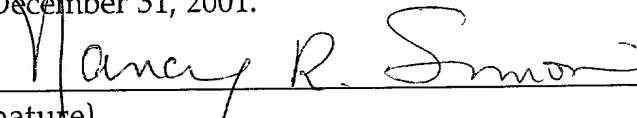
APPLICATION FOR  
UNITED STATES UTILITY LETTERS PATENT

Be it known that we, David V. James and William Rivard, have  
invented certain new and useful improvements in an

APPARATUS AND METHOD FOR INTER-NODE  
COMMUNICATION

of which the following is the Specification:

I hereby certify that this correspondence is being  
deposited with the United States Postal Service as  
Express Mail No. EK889946752US in an envelope  
addressed to: BOX PATENT APPLICATION,  
Commissioner of PatentS, Washington, D.C. 20231,  
on December 31, 2001.

  
(Signature)

## CROSS REFERENCE TO RELATED APPLICATION

This is a division of United States Patent Application No. 09/040,149, filed on March 17, 1998.

## BACKGROUND

## 1. Technical Field

The present invention relates generally to apparatus and methods for transmitting signals between nodes, and more particularly for transmitting signals at high bit-rates between nodes.

## 2. Background

A typical computer or inter-linked set of computers can be modeled as a series of nodes which communicate with one another point-to-point. Although nodes have in the past been attached to a bus, modern communications standards more commonly employ point-to-point interconnections. In the past, the communication data rate between such nodes was limited more by the performance of the computer or its various internal chips than by the speed of the traces or transmission lines by which the nodes were connected. Now as chip and computer speeds have substantially increased, these interconnects are hindering further performance improvements.

More specifically, current signaling architectures as well as the physical limitations of the traces and transmission lines themselves limit the maximum inter-node communications rate. Synchronized buses and point-to-point links are two of interconnect architectures commonly used. Synchronous bus architectures typically broadcast an address block to all nodes on a multiplexed bus. The node corresponding to the address then generates an acknowledgement block, which is also broadcast to all of the nodes. This architecture results in relatively low communications

1 throughput. This is because each node must be synchronized to the same  
2 common reference clock so that address and acknowledgment blocks can be  
3 transmitted and received. Nodes employing the synchronous bus  
4 architecture must also take turns communicating, further limiting the  
5 maximum possible inter-node data rate, especially when a large number of  
6 nodes are connected to the same bus. The multiplexed buses used with  
7 synchronous bus architectures also typically contain intermediate stubs and  
8 additional signal paths that limit the effective speed of data transfer between  
9 nodes.

10  
11 Point-to-point link architectures are comparatively more time efficient  
12 since their signals have an affiliated reference clock signal. Using a  
13 point-to-point link architecture, two nodes can transfer data at a rate  
14 independent of other nodes and of any common reference clock. However,  
15 point-to-point link inter-node data rates still tend to be limited by the physical  
16 limitations in the traces and transmission lines.

17  
18 For instance, modern computers typically communicate by either  
19 single-ended or differential signaling. Both of these forms of signaling are  
20 well known in the art. Ideally, single-ended connections require only one  
21 physical line per logic signal. However, as communication rates have  
22 increased so has ground-bounce, which is inherent in single-ended systems.  
23 Attempts to solve the ground-bounce problem include adding power supply  
24 and ground pins for each single-ended logic line on a chip, effectively tripling  
25 the number of physical traces required. Thus, six single-ended logic signals  
26 can require up to eighteen physical traces. Differential signaling systems  
27 require two physical traces for each logic signal. Thus, six differential logic  
28 signals require at least twelve physical traces. Since silicon and computer  
29 resources are finite, a large number of traces or transmission lines can  
30 significantly increase the cost of manufacturing the chip or the computer.

1        Regardless of whether single-ended or differential signaling is used, the  
2 physical traces and transmission lines all have an inherent parasitic  
3 inductance. As the data rate over these pathways increases, the parasitic  
4 inductance combined with the quickly varying signal currents generate  
5 parasitic voltages that interfere with and corrupt the signals traveling over  
6 these pathways.

7  
8        Additionally, large signal currents that pass through the traces and  
9 transmission lines can generate Electro-Magnetic Interference (EMI) noise  
10 which further corrupts signals traveling between the nodes. Such EMI noise  
11 may also, from time to time, exceed the limits of various well known  
12 regulatory standards for permissible EMI radiation levels.

13  
14        Other prior art approaches have employed RAMBUS technology  
15 (manufactured by RAMBUS, Inc. of Mountain View, California) to reduce the  
16 parasitic and EMI noise voltages present on some signal lines. The RAMBUS  
17 approach consists of a number of traces or transmission lines, each of which  
18 transmits a different signal. Ideally, these signal lines are kept in close  
19 proximity to one another. One of the signal lines is designated as a reference  
20 and used to cancel out some of the noise effects present on the signal lines. A  
21 shortcoming of this approach is a noticeable current surge when all of the  
22 signal lines are either logic 1's or logic 0's.

## SUMMARY

The present invention delineates an inter-node communications paradigm for enabling signals to be transmitted between nodes at a higher rate. The higher rate is possible due to an encoding schema that reduces current demands and fluctuations between multiple nodes. The encoding schema also requires fewer physical traces and/or transmission lines than high speed single-ended and differential signaling circuits.

Within the apparatus of the present invention, a first node is connected to a communication channel. Operations on the first node result in a first set of signals that are to be transmitted over the communication channel. The logic states which comprise the first set of signals may range from all logic zeros to all logic ones. This large number of potential logic transitions results in large current fluctuations over the communication channel. To reduce and/or eliminate these current fluctuations, the present invention also includes an encoder or lookup table for transforming the first set of signals into a second set of signals having either an equal number, nearly an equal number, a constant number, or nearly a constant number of logic ones and logic zeros. In one embodiment, groups of six signals from the first set of signals are encoded into eight signals in the second set of signals.

Within the method of the present invention, a first set of signals from a first node are encoded into a second set of signals having either an equal number, nearly an equal number, a constant number, or nearly a constant number of logic ones and logic zeros. This second set of signals is then transmitted over a communication channel.

Thus, the present invention presents a communications technique which has quieter switching currents than single-ended circuits and requires fewer physical traces and transmission lines than differential circuits. The

1 present invention can be applied to communications between computer chips  
2 on a circuit board as well as between nearby computers linked together. These  
3 and other aspects of the invention will be recognized by those skilled in the  
4 art upon review of the detailed description, drawings, and claims set forth  
5 below.

10040156-10040101

BRIEF DESCRIPTION OF THE DRAWINGS

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19

Figure 1 is a block diagram of an apparatus for inter-node communication;

Figure 2 is a table of unencoded signals utilized by each node within the apparatus of Figure 1;

Figure 3 is a block diagram of an apparatus for encoding and transmitting signals between a set of nodes;

Figure 4 is a partial electrical circuit of an encoder and decoder pair within the apparatus for encoding and transmitting the signals between the set of nodes;

Figure 5 is a flowchart of a method for inter-node communication; and

Figure 6 is a flowchart of a method for encoding inter-node communication signals.

## DETAILED DESCRIPTION

**Figure 1** is a block diagram of an apparatus 100 for transmitting high bit-rate signals between a set of nodes. The apparatus 100 includes a first node 102, a second node 104, a third node 106, a fourth node 108 and a clock generator 110. The apparatus is designed to provide the functionality of a synchronous bus, with the performance advantages of point-to-point signaling. Each node 102, 104, 106 and 108 receives a clock signal on line 112 from the clock generator 110, data signals on lines 114 A, B, C and D, flag signals on lines 116 A, B, C and D, and strobe signals on lines 118 A, B, C and D. Collectively, the signals on lines 114, 116, and 118 make up a shared communications channel/link between the nodes 102, 104, 106 and 108. Those skilled in the art will be aware of various other receive and transmit signals which can also be passed between the nodes 102, 104, 106 and 108, depending upon the application or transmission protocol.

Data transmissions between nodes 102, 104, 106 and 108 are source synchronous. Even though all of the nodes also receive the clock signal on line 112, the strobe signals on lines 118 A, B, C and D act as a reference clock for these data transmissions. The strobe signal resolves any frequency differences in timing between the transmitting and receiving node. Although each node has to compensate for arbitrary frequency differences between its incoming data and the clock signal on line 112, this arrangement eliminates the need to insert or delete symbols to adjust for inter-node timing differences. Strobe signals also enable higher transmission bandwidths between the nodes because there is no longer a need to accurately synchronize all of the nodes within the apparatus 100. The inter-node transmission rates are not affected by inter-node transmission delays, thus arbitrary delays between each of the nodes are acceptable.



1       Using the point-to-point link data communication architecture, the  
2 nodes transmit requests and responses as *send* packets. The receiving node  
3 strips the *send* packet of its data and then returns a small *acknowledgment*  
4 (i.e. "*ack*") packet. Routing architectures for the packets are preferably  
5 straightforward. In the preferred embodiment, the *send* packets addressed to  
6 a specific node are stripped from the inter-node data stream flowing on lines  
7 114 A, B, C, and D by that node and are replaced with *ack* packets. The *send*  
8 and *ack* packets addressed to other nodes are passed through any intermediate  
9 nodes. The *ack* packets are stripped from the inter-node data stream when  
10 they return to the original transmitting node.

11  
12       For example, in order for node 102 to transmit data to node 108, node  
13 102 must first generate a *send* packet addressed to node 108. Node 102 then  
14 transmits this *send* packet to node 106. Node 106 looks at the address of the  
15 *send* packet and since the *send* packet is not intended for node 106, passes the  
16 *send* packet along to node 108. Node 108 then looks at the address of the *send*  
17 packet and since the *send* packet is intended for node 108, strips the *send*  
18 packet from the inter-node data stream and generates an *ack* packet addressed  
19 to node 102. Node 108 then sends the *ack* packet to node 104. Node 104 looks  
20 at the address of the *ack* packet and since the *ack* packet is not intended for  
21 node 104, passes the *ack* packet along to node 102. Node 102 then looks at the  
22 address of the *ack* packet and since the *ack* packet is intended for node 102,  
23 strips the *ack* packet from the inter-node data stream. After this last step the  
24 data exchange between nodes 102 and 108 is complete.

25  
26       The flags on lines 116 A, B, C, and D are used for data stream packet-  
27 framing. Data stream packet-framing consists of labeling each packet with  
28 either a first-packet symbol, a between-packet-idle symbol, or a last-packet  
29 symbol. These flags are also used for arbitration purposes. Arbitration  
30 protocols place a limit on a number of consecutive packets that can be sent by

1 any one node. This ensures that each node has an opportunity to send its  
2 packets over the shared link.

3  
4 Preferably one, and only one, of the nodes 102, 104, 106, 108 functions as  
5 a "scrubber" within the apparatus 100. The scrubber performs such  
6 maintenance functions as, removing mis-addressed packets on their second  
7 pass around the shared link. The scrubber can be selected by setting a *scrubId*  
8 line 120 A, B, C or D on the selected node to logic "1" (node 104 in Figure 1)  
9 and setting the remaining *scrubId* lines 120 A, B, C or D to logic "0" (nodes 102,  
10 106 and 108 in Figure 1). Those skilled in the art will know of other vendor-  
11 dependent scrubber-assignment techniques that can be used.

12  
13 **Figure 2** is a table 200 of unencoded signals utilized by each node  
14 within the apparatus of **Figure 1**. The signals include a 1-pin scrubber  
15 identifier (*scrubId*) 202, a 1-pin central clock (clock) 204, a 1-pin strobe 206,  
16 4-pins of flags 208, and 32-pins of data.

17  
18 The *scrubId* signal 202 is received by the nodes on lines 120 A, B, C, and  
19 D, and uniquely identifies a particular node as the scrubber within the shared  
20 communications channel (also known by those skilled in the art as a ringlet).  
21 In an alternate embodiment, a vendor-dependent scrubber-identification  
22 technique can be provided.

23  
24 The clock signal 204 is preferably an input-only signal received by the  
25 nodes on line 112, and provides the nodes with a reference for synchronizing  
26 their internal frequency-lock loops. The frequency-lock-loops within the  
27 nodes permit a lower clock generator 110 frequency, which in turn simplifies  
28 clock signal distribution. For example, a 50 MHz clock signal can be  
29 frequency-lock-looped up to 500 MHz. Also since the frequency-lock loops  
30 within the nodes are tracking the clock signal fairly closely, the cycles per  
31 second each node sees is about the same.

1  
2       Frequency locking, rather than phase locking, is used because arbitrary,  
3 but fixed, phase differences can be tolerated and a small input-data FIFO can  
4 compensate for any incoming phase differences. The FIFO compensates for  
5 the fixed phase error between each node. The fixed phase error is random  
6 and cannot be controlled.

7  
8       The strobe signal 206, received by the nodes on lines 118 A, B, C, and D,  
9 is preferably complemented on each cycle, so that a receiving node can  
10 accurately determine when incoming (source-synchronous) data should be  
11 latched.

12  
13       The flag signals 208, received by the nodes on lines 116 A, B, C, and D,  
14 transfer control and framing information between the nodes.

15  
16       The data signals 210, received by the nodes on lines 114 A, B, C, and D,  
17 transmit the contents of the *send* and *ack* packets. Preferably, most-significant  
18 bits are sent first when sending a packet header, and lower addresses are sent  
19 first when sending data. While the present inter-node data packets contain 32  
20 data bits, other data packet capacities, such as 8, 16, 64, or 128 data bits, are also  
21 acceptable. Alternatively, large data-words may be broken up and sent over  
22 multiple clock cycles. For instance a large 64 bit data packet could be sent as  
23 two smaller 32 bit data packets. When deciding how many bits to include in a  
24 data packet, designers should consider that while 8-bit data packets may be  
25 more cost effective in terms of coding or hardware to implement, such small  
26 data packet designs have less bandwidth per pin due to the relatively-fixed  
27 overhead of the scrubId, clock, strobe, and flag pins. However, while 128-bit  
28 data packets may have more bandwidth per pin, such large data packets may  
29 be more expensive to implement in terms of coding, hardware, and/or  
30 skew-management circuits.

1       **Figure 3** is a block diagram of an apparatus 300 for encoding and  
2 transmitting the signals 206, 208, and 210 between a set of nodes. The  
3 apparatus includes a driver node 302 and a receiver node 304 which can  
4 represent any pair of the **Figure 1** nodes 102, 104, 106, and 108. The driver  
5 node 302 includes a plurality of encoders 302 A-G and the receiver node 304  
6 includes a plurality of decoders 304 A-G. The nodes 302 and 304 communicate  
7 via a set of traces, circuit paths, and/or transmission lines 308. These traces,  
8 circuit paths, and transmission lines 308 collectively make up part of the  
9 communications channel between the two nodes and perform the same roles  
10 as the data 114, flag 116, and strobe 118 lines of **Figure 1**. The nodes  
11 themselves can include any number of inter-linked devices. These  
12 inter-linked devices can be computer chips, circuit boards, and/or stand alone  
13 computers.

14  
15       The apparatus 300 transmits the signals between the nodes by either a  
16 dedicated line (such as for the scrubId signals on lines 120 A-D and the clock  
17 signal on line 112) or after implementing an encoding schema (such as for the  
18 strobe, flags and data signals). By selectively encoding the strobe, flag and data  
19 signals transmitted between the nodes 302, 304, ground-bounce during  
20 high-speed signaling, can be reduced. Preferably the strobe, flag and data  
21 signals are grouped and encoded in such a way that nearly an equal (called  
22 DC-free encoding) and/or constant (called DC-balanced encoding) number of  
23 logic 0's and 1's are always transmitted between the driver node 302 and the  
24 receiver node 304. These encoding schemas are preferably implemented  
25 using an even number of data lines 308.

26  
27       One way to implement a DC-balanced encoding schema is shown in  
28 **Figure 3**. The strobe 206 signal is fed into a 1 to 2 encoder 302A which uses a  
29 complementary encoding schema before the signal is transmitted over the  
30 data lines 308. The flag 208 and data 210 signals, however, are divided into  
31 groups of six unencoded signals (i.e. 6-bits), which are then converted into

1 groups of eight encoded signals (i.e. 8-bits). These eight encoded signals are  
2 then transmitted in parallel over the data lines 308. Other signal coding  
3 schemas, such as when groups of four unencoded signals (i.e. 4-bits) are  
4 converted into groups of six encoded signals (i.e. 6-bits), can also be used.

5  
6 DC-free encoding schemas can be implemented by transmitting an  
7 even number of encoded signals with an equal number of logic 1's and 0's in  
8 parallel over the data lines 308.

9  
10 The DC-Balanced and DC-Free encoding schemas have the following  
11 characteristics: First, a constant number of 1's valued lines is always driven  
12 and thus the driver node 302 is balanced. A balanced driver node means that  
13 total current over the data lines 308 is fairly constant. Second, ground-bounce  
14 noise is reduced, since logic transitions from all zeros to all ones and from all  
15 ones to all zeros are eliminated. Third, an implied reference voltage can be  
16 obtained by averaging all of the data line 308 voltages. Fourth, parity  
17 protection is inherent, since all single-bit transmissions failures, and many  
18 double-bit errors, can be detected as an illegal (i.e. non-DC-balanced) input  
19 code value. Fifth, peak current demands are reduced, since for each of the  
20 unencoded signals, only half of the data lines are actively driven to a high  
21 current logic state (such as logic "1"). Sixth, the encoding schema allows extra  
22 control characters to be transmitted (for instance, in a 6-to-8 bit encoding  
23 schema there are 6 unmapped 8-bit encoded values for each set of 64  
24 unencoded values).

25  
26 A nearly DC-free/nearly DC-balanced encoding schema can be  
27 implemented by transmitting an odd number of bits, which contain no more  
28 than one extra logic 1 or logic 0, in parallel over the data lines 308. For  
29 instance, a 6/7 encoding schema (i.e. where 6-bits are encoded into 7-bits) may  
30 be used where a logic 1 to logic 0 ratio is either 3-to-4 or 4-to-3. Although  
31 more efficient, this nearly free/balanced encoding schema has no parity

1 protection and may be subject to signal-integrity limitations. This encoding  
2 schema also results in a less accurate threshold reference voltage, once all of  
3 the received signal values are averaged by the decoders 304 A-G.

4  
5 The encoding schema herein taught requires extra-pinouts (for  
6 example, the 6-to-8 schema requires an additional 2-pins for every 6 signals).  
7 However, these additional demands upon limited silicon and PC-board  
8 resources are offset by a reduction in a number of required ground and/or  
9 power pins when compared with current unencoded full-swing signals  
10 transmitted over single-ended data lines. To localize any chip or PC-board  
11 ground-plane currents, each set of 8 signal traces is preferably routed as a  
12 group. While the 6-to-8 bit encoding schema is preferred, sometimes the  
13 number of unencoded signal lines are not always multiples of 6. In those  
14 cases other encoding options are possible, such as 1-to-2 (differential), 2-to-4  
15 (also differential), and 4-to-6 encoding schemas.

16  
17 **Figure 4** is a partial electrical circuit 400 of the encoder 302A and the  
18 decoder 304A pair within the apparatus 300 for encoding and transmitting the  
19 signals between the set of nodes. The circuit 400, shown in **Figure 4**, effects a  
20 6-to-8 bit encoding schema which employs a simple technology-independent  
21 technique for driving the data lines 308 by switching a constant current 402  
22 (i.e. "4i") to half (i.e. 4) of the total number (i.e. 8) of data lines 308 which are  
23 driven to a high current logic state lines. Only half of the data lines are  
24 driven since a DC-free encoding schema is effected. A zero volt termination  
25 voltage 404, rather than a fixed positive voltage, enables the circuit to be  
26 supply-voltage independent. In the circuit 400, termination resistors,  $R_t$ , are  
27 effected using on chip FETs. The value of  $R_t$  is matched to the impedance of  
28 each data-line 308. A typical termination resistance is  $75\Omega$ . Averaging  
29 resistors,  $R_c$ , are chosen so that the threshold reference voltage of the decoder  
30 304A is one half of the driven signal levels on the data lines 308. The decoder  
31 304A then uses differential amplifiers 406 to compare the signals on the

1 data-lines 308 with the threshold reference voltage to determine whether, for  
2 example, a logic "1" or a logic "0" has been transmitted.

3  
4 Portions of the encoding and decoding hardware described in **Figure 3**  
5 may also be implemented in software. In so doing, additional configurable  
6 elements, such as a processing unit, a memory, and a storage device (not  
7 shown) need to be added to **Figures 1, 3, and 4**. The memory would store  
8 computer program instructions for controlling how the processing unit  
9 accesses, transforms and outputs data. Those skilled in the art will recognize  
10 that in alternate embodiments the memory could be replaced with a  
11 functionally equivalent computer-useable medium such as a compact disk, a  
12 hard drive or a memory card.

13  
14 **Figure 5** is a flowchart of a method for inter-node communication.  
15 The method begins in step 502 where a driver node and a receiver node are  
16 identified within a communications system. Next, in step 504, a set of  
17 unencoded signals transmitted from the driver node to the receiver node are  
18 identified. The signals include a set of data values (i.e. logic "1's" and logic  
19 "0's"). In step 506, the unencoded data values of the unencoded signals are  
20 encoded to produce encoded signals. Next, in step 508, the encoded signals are  
21 transmitted from the driver node to the receiver node. In step 510, the  
22 encoded signals are decoded. After step 510, the method for inter-node  
23 communication is complete.

24  
25 **Figure 6** is a flowchart of a method for encoding the data values of the  
26 inter-node communication signals (step 506 of **Figure 5**). The method begins  
27 in step 602 by selecting a code such that a difference between a total number of  
28 unencoded data values and a total number of encoded data values is a small  
29 predetermined fraction of the total number of unencoded data values. A 6-bit  
30 unencoded signal to 8-bit encoded signal coding schema is preferred, however  
31 a 4-bit unencoded signal to 6-bit encoded signal coding schema may be more

1 appropriate in some applications. Codes may be stored in a set of lookup  
2 tables or generated using algorithmic methods. Codes may also be switched at  
3 any point during signal transmission provided that the switch is  
4 communicated to the receiver node. In step 604, the code is selected such that  
5 an encoded signal has an equal number of logic 1's and 0's. Alternatively, in  
6 step 606 the code is selected such that an encoded signal has nearly an equal  
7 number of logic 1's and 0's. Alternatively, in step 608 the code is selected such  
8 that an encoded signal has a constant number of logic 1's and 0's.  
9 Alternatively, in step 610 the code is selected such that an encoded signal has  
10 nearly a constant number of logic 1's and 0's. After step 610, the method for  
11 encoding the inter-node communication signals is complete.

12  
13 While the present invention has been described with reference to a preferred  
14 embodiment, those skilled in the art will recognize that various  
15 modifications may be made. Variations upon and modifications to the  
16 preferred embodiment are provided by the present invention, which is  
17 limited only by the following claims.